METHODS AND SYSTEMS FOR GENERATING TEXTUAL INFORMATION

RELATED APPLICATIONS

[0001] This application relates to Attorney Docket No. GP-175-36-US, filed herewith and entitled, "Methods and Systems for Processing Textual Information," the entirety of which is incorporated herein by reference.

FIELD OF THE INVENTION

[0002] The invention generally relates to methods and systems for generating textual information.

BACKGROUND

[0003] As World Wide Web ("web") search engines (sometimes referred to as "Internet Search Engines") have improved, many users have turned to these search engines for navigating the web, rather than inputting uniform resource locators (URLs) into browser address fields or using browser bookmarks. Search engines may perform searches of various databases, which may be public, e.g., the Internet, and/or private, e.g., an intranet, a client device, etc., using one or more known search techniques. For example, one known search technique, described in an article entitled, "The Anatomy of a Large-Scale Hypertextual Search Engine," by Sergey Brin and Lawrence Page, assigns a degree of importance to a document, such as a web page, based on the link structure of the web.

[0004] As efficient as a search engine may be, its value to a user may be limited by the manner in which the search engine provides a summary of search results to a user. For instance, search engines generally provide summaries (sometimes referred to as "snippets") of documents or websites located in response to a query. A user browses such summaries, and typically selects a link associated with a summary that best matches the search criteria to view the entire document or to navigate to the desired web page. Summaries that provide too much information can consume output (e.g., display) resources and can overwhelm a user with extraneous information, which can slow down the user's search. Summaries that provide too little information may not provide the user with sufficient information to identify relevant documents. In either case, such summaries are generally ineffective in aiding a user's search for desired information.

SUMMARY

[0005] Embodiments of the present invention comprise methods and systems for generating textual information. In one exemplary embodiment, a method of generating textual information is disclosed that comprises identifying a plurality of candidate summaries related to textual information based, at least in part, on a document, determining first and second attribute values based, at least in part, on the candidate summaries, and determining an optimal candidate summary based at least in part on the first and second attribute values.

[0006] In another exemplary embodiment, a method comprises searching a document, identifying a keyword disposed in the document, identifying a plurality of candidate summaries related to textual information based, at least in part, on the document, determining a number of storage locations for the plurality of candidate summaries, combining the plurality of candidate summaries into a plurality of combined candidate summaries, determining first and second attribute values based, at least in part, on the candidate summaries, selecting from the plurality of combined candidate summaries a first highest-weighted combined candidate summary and a second highest-weighted combined candidate summary, determining an optimal candidate summary based at least in part on the first and second attribute values, and comparing the first and second highest-weighted combined candidate summaries. In one embodiment the number of storage locations can be based, at least in part, on a size of the document. In another embodiment, determining the optimal candidate summary may be further based at least in part on the comparison of the first and second highest-weighted combined candidate summaries.

[0007] These exemplary embodiments are mentioned not to limit or define the invention, but to provide examples of embodiments of the invention to aid understanding thereof. Exemplary embodiments are discussed in the Detailed Description, and further description of the invention is provided there. Advantages offered by the various embodiments of the present invention may be understood by examining this specification.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The accompanying drawings, which constitute part of this specification, help to illustrate embodiments of the invention.

[0009] Figure 1 is a block diagram of an exemplary embodiment for implementing an embodiment of the present invention.

[0010] Figure 2 is a block diagram of a method according to an embodiment of the present invention.

DETAILED DESCRIPTION

[0011] Embodiments of the present invention include methods and systems for generating textual information. Exemplary embodiments are described herein.

System Architecture

[0012] Referring now to the drawings in which like numerals indicate like elements throughout the several figures, Figure 1 is a block diagram illustrating an exemplary embodiment of the present invention. The system 100 shown in Figure 1 includes multiple client devices 102a-n in communication with a server device 104 over a network 106. The network 106 shown comprises the Internet. In other embodiments, other networks, such as an intranet may be used. Moreover, methods according to the present invention may operate within a single computer.

[0013] The client devices 102a-n shown each include a computer-readable medium, such as a random access memory (RAM) 108, coupled to a processor 110. The processor 110 executes computer-executable program instructions stored in memory 108. Such processors may include a microprocessor, an ASIC, and state machines. Such processors include, or may be in communication with, media, for example computer-readable media, which stores instructions that, when executed by the processor, cause the processor to perform the steps described herein. Embodiments of computer-readable media include, but are not limited to, an electronic, optical, magnetic, or other storage or transmission device capable of providing a processor, such as the processor 110 of client 102a, with computer-readable instructions. Other examples of suitable media include, but are not limited to, a floppy disk, CD-ROM, DVD, magnetic disk, memory chip, ROM, RAM, an ASIC, a configured processor, all optical media, all magnetic tape or other magnetic media, or any other medium from which a computer processor can read instructions. Also, various other forms of computer-readable media may transmit or carry instructions to a computer, including a router, private or public network, or other transmission device or channel, both wired and wireless. The instructions may comprise code from any suitable computer-programming language, including, for example, C, C++, C#, Visual Basic, Java, Python, Perl, and JavaScript.

[0014] Client devices 102a-n may also include a number of external or internal devices such as a mouse, a CD-ROM, DVD, a keyboard, a display, or other input or output devices. Examples of client devices 102a-n are personal computers, digital assistants, personal digital assistants, cellular phones, mobile phones, smart phones, pagers, digital tablets, laptop computers, Internet appliances, and other processor-based devices. In general, a client device 102a may be any suitable type of processor-based platform that is connected to a network 106 and that interacts with one or more application programs. Client devices 102a-n may operate on any operating system capable of supporting a browser or browser-enabled application, such as Microsoft® Windows® or Linux. The client devices 102a-n shown include, for example, personal computers executing a browser application program such as

Microsoft Corporation's Internet Explorer™, Netscape Communication Corporation's Netscape Navigator™, and Apple Computer, Inc.'s Safari™.

[0015] Through the client devices 102a-n, users 112a-n can communicate over the network 106 with each other and with other systems and devices coupled to the network 106. As shown in Figure 1, a server device 104 is also coupled to the network 106. The server device comprises a server processor 116 and a server memory storage device 118. The server device 104 shown comprises a single computer. However, in other embodiments, the server device 104 may comprise multiple and/or distributed devices, or there may be no server device.

[0016] In one embodiment, a user 112a-n generates a search query 114 at a client device 102a. The client device 102a transmits the query 114 to the server device 104 via the network 106. For example, a user 112a types a textual search query into a query field of a web page of a search engine interface or other client-side software displayed on the client device 102a, which is then transmitted via the network 106 to the server device 104.

[0017] In the embodiment shown, a user 112a inputs a search query 114 at a client device 102a, which transmits an associated search query signal 122 reflecting the search query 114 to the server device 104. The search query 114 may be transmitted directly to the server device 104 as shown. In another embodiment, the query signal 122 may instead be sent to a proxy server (not shown), which then transmits the query signal 122 to server device 104. Other configurations are possible.

[0018] The server device 104 shown includes a server executing a search engine application program, such as the Google™ search engine. Similar to the client devices 102a-n, the server device 104 shown includes a processor 116 coupled to a computer-readable memory 118. Server device 104, depicted as a single computer system, may be implemented as a network of computer processors. Examples of a server device 104 are servers, mainframe computers, networked computers, a processor-based device, and similar types of systems and devices. Client processor 110 and the server processor 116 can be any of a number of computer processors, such as processors from Intel Corporation of Santa Clara, California and Motorola Corporation of Schaumburg, Illinois.

[0019] Memory 118 contains the search engine application program, also known as a search engine 120. The search engine 120 locates relevant information in response to a search query 114 from a user 112a-n. The search engine 120 then provides the result set 124 to the client 102a via the network 106.

[0020] In the embodiment shown, the server device 104, or related device, has previously performed a crawl of the network 106 to locate articles, such as web pages, stored at other devices or systems connected to the network 106, and indexed the articles in memory 118 or on another data storage device. Articles include, for example, web pages of various formats, such as HTML, XML, XHTML, Portable Document Format (PDF) files, and word processor, database, and application program document files, audio, video, or any other documents or information of any type whatsoever made available on a network (such as the Internet), a personal computer, or other computing or storage means. The embodiments described herein are described generally in relation to HTML files or documents, but embodiments may operate on any type of article, including any type of image.

[0021] It should be noted that various embodiments of the present invention may comprise systems having different architecture than that which is shown in Figure 1. For example, in some systems according to the present invention, server device 104 may comprise a single physical or logical server, or there may be no server. The system 100 shown in Figure 1 is merely exemplary, and is used to explain the exemplary method shown in Figure 2.

Process And Example

[0022] Various methods may be implemented in the environment shown in Figure 1 and other embodiments, according to the present invention. Methods according to the present invention may be implemented by, for example, a processor-executable program code stored on a computer readable medium.

[0023] Referring now to Figure 2, a method 200 according to one embodiment of the present invention is shown. The method 200 may be used to enhance use of and navigation through a retrieved collection of search results which may identify documents or websites. The method 200 may be further used to refine a query such that the desired information may be obtained more precisely. For example, the method 200 may be used to generate a candidate summary of a document, website, or other information, which may include textual information.

[0024] An example of the method 200 comprises examining and processing a document returned as part of a search-result set to generate a candidate summary of the document. The method 200 will be described below with reference to an example, the purpose of which is to aid understanding of the present invention. Thus, it is to be understood that the present invention is not limited to the example described below.

[0025] As indicated by block 210, the method 200 comprises determining a candidate summary size. As described above, a lengthy document summary may provide so much information to the user 112a, that the user 112a cannot effectively and/or efficiently process the generated information. Thus, a desired size of the summary can be predetermined. In one embodiment, the user 112a requests a result set 124 to include a summary or summaries not to exceed a desired length of a character string, for example, a character string of 30, 40, 50, or 60 characters, etc. In another embodiment, the length of the summary character string can be determined dynamically by the processor. For purposes of the present example, the summary length is 40 characters.

[0026] In one embodiment, the user 112a requests a result set 124 to include a summary or summaries not to exceed a specified number of pixels on a display. This can be useful when the available space for the summary is relatively small and/or is desirable that the summary fits in a specific number of lines in the display. The number of pixels may be determined based at least in part on the size of the display. For example, if results are displayed in a browser, a length of the candidate summary may be generated appropriate for the size of the browser window (e.g., the candidate summary occupies a specific number of lines of the display).

[0027] As indicated by block 215, the method 200 comprises receiving a search request. In the present example of the method 200, a user 112a generally submits a search query 114 for a particular document or documents that include a desired keyword included in the query (a keyword is defined broadly herein to mean one or more terms, such as words, acronyms, or other character strings). The query 114 is communicated as a query signal 122 over the network 106 to server device 104. In one embodiment, a summary may be generated for results obtained via automatic implicit queries.

[0028] For example, the keyword may comprise a particular search term or string of search terms. In one embodiment, a keyword is communicated by the query signal 122 to the processor 116. The processor 116 identifies keywords from an input string disposed in the document and/or from the list of tokens, i.e., the tokenized document.

[0029] As indicated by block 220, the method 200 comprises identifying a plurality of documents. The documents may include documents located on a dedicated server, a local network, or the internet. In an embodiment, the document comprises a number of character strings. In another embodiment, the document comprises a number of words and/or other textual information. For example, the documents may include webpages or other textual information. Generally, the documents are searched for the keyword query, and documents that include the keyword are identified by the server device 104.

[0030] As indicated by block 225, the method 225 comprises selecting a document for operation. In one embodiment, documents may be selected by the presence of one or more keywords in the document. In another embodiment, documents may be selected by a relative weighting or scoring system, such as, for example the number of occurrences of the keyword in the document. Other suitable methods of selecting documents can be used. In one embodiment, a plurality of text documents may be retrieved by the search engine 120 in response to the query signal 122, and operated upon.

[0031] As indicated by block 230, the method 200 comprises tokenizing a document. To manipulate and process a document, it may be necessary to tokenize, e.g., parse, the document into character strings or words. As used herein, tokenizing generally refers to separating an input string into a list of tokens, such as for example, contiguous character strings representing a plurality of individual words. In an embodiment, an input string is separated on "white space," i.e., gaps or breaks, between individual characters. The input string can also be separated on punctuation or other non-alphanumeric characters. In one embodiment, multiple words may be treated as a single token, such as for example, a string enclosed in double quotes, proper names, dates, and times. Alternatively, other suitable rules for tokenizing can be used.

[0032] As indicated by block 235, the method 200 comprises determining a number of storage locations for candidate summaries related to textual information based, at least in part, on the document. For example, the storage locations may be used to store data retrieved or extracted from the document or documents for further processing and/or manipulation. In one embodiment, each of the storage locations comprises a bucket. In an embodiment, a bucket refers to a temporary or permanent data repository, such as for example, a memory cache.

[0033] In one embodiment, the number of storage locations is based, at least in part, on at least one of a size of a document, a desired size of the candidate summary, and a number of query terms. A larger document generally will include a greater number of storage locations than a comparatively smaller document. In one embodiment, the number of storage locations is also based, at least in part, on the number of keywords in the search query, so that a query with more keywords will generate greater number of storage locations for candidate summaries. In another embodiment, the number of storage locations is limited to a predetermined number regardless of the size of the document. In the present example, the number of storage locations is limited to 20 buckets.

[0034] As indicated by block 240, the method 200 comprises identifying a plurality of candidate summaries related to textual information based, at least in part, on the document. In one embodiment, the candidate summaries comprise snippets of text or textual information. Methods of generating candidates are described in co-pending application (Attorney Docket No. GP-175-36-US), the entirety of which is incorporated herein by reference. In one embodiment, the document is tokenized until a predetermined number of candidate summaries is generated. In another embodiment, the number of candidate summaries is dynamically determined.

[0035] As will be described in further detail below, the candidate summaries can be used to generate a summary of the document. In one embodiment, the candidate summaries comprise candidate text or other data for possible placement in a summary of a document. The plurality of candidate summaries can be, for example, textual information based, at least in part, on a document. In one embodiment, textual information based, at least in part, on a document comprises a keyword found in a document and a character string disposed proximate to the keyword. The character string may include an entire sentence or portions of the sentence including the keyword. The character string may also include a predetermined number of words or characters disposed before and after the keyword. In another embodiment, the candidate summaries comprise one or more keywords.

[0036] The character string can comprise a first character string and a second character string. In an embodiment, the first character string comprises a first number of words (e.g., five, ten, fifteen, etc.) and the second character string comprises a second number of words (e.g., five, ten, fifteen, etc.). Generally, the first and second character strings can be disposed on opposing ends of the keyword.

[0037] As indicated by block 245, the method 200 shown in Figure 2 comprises determining first and second attribute values based, at least in part, on the candidate summaries. Generally, the first and second attribute values can be used to generate the document summary. The use of the first and second attribute values to generate the document summary will be described in further detail below. In one embodiment, the first and second attribute values may correspond to a total number of words, and can be mapped onto the bucket. For example, the number of words comprises a total number of words in the first character string, the keyword, and the second character string.

[0038] In one embodiment, a user 112a selects the first and second attribute values. For example, the user 112a may specify a first attribute value of 40 characters and a second attribute value of 45 characters. In another embodiment, the first and second attribute values are automatically generated by, for example, the processor 116. In one embodiment, the second attribute value may be greater than the first attribute value. In one embodiment, the size of the document summary is limited to the second attribute value. In the present example, the first attribute value is 40 characters and the second attribute value is 50 characters.

[0039] As described above, the number of characters provided herein are presented for the sake of example only. Other suitable numbers of characters can be used. Furthermore, different numbers of characters can be used for different categories of documents, such as, for example, web results and e-mails. Moreover attribute values can also be based, at least in part, on pixel width and number of lines.

[0040] In another embodiment, the first and second attribute values comprise a pixel size. Certain string characters occupy less space than other characters in many fonts. For example, in many fonts, the letter "i" is fewer pixels wide than the letters "m" or "w" and thus occupies less space. Therefore, two storage locations, each having a capacity of 25 pixels, may hold a different number of words depending on the pixel size of the words in a particular character string. Alternatively, other suitable attribute values can be used.

[0041] As indicated by block 250, the method 200 comprises combining the plurality of candidate summaries into a plurality of combined candidate summaries. The combined candidate summaries can form a document summary. In one embodiment, a first combined candidate summary does not exceed the first attribute value (in the present example 40 characters) and a second combined candidate summary does not exceed the second attribute value (in the present example 50 characters).

[0042] In the present example, the processor 116 determines or identifies the highest-weighted candidate having a length of five characters. The processor 116 repeats this process in increments of five characters, until the highest weighted 40-character string is identified. Of course, other suitable increments can be used. In one embodiment, the increment is based at least in part on a maximum number of desired characters in the candidate summary.

[0043] The processor 116, combines the plurality of identified candidates in the predetermined increments such that a first set of combined candidates does not exceed the total number of desired characters. For example, when the total number of desired characters is 40 characters, the highest-weighted 10-character candidate summary is combined with the highest-weighted 30-character candidate summary and the highest-weighted 15-character candidate summary is combined with the highest-weighted 25-character candidate summary.

[0044] In the present example, the group of 40-character candidates (i.e., the first set) combined in this manner are compared to one another and the processor 116 selects the highest-weighted 40-character candidate (i.e., the first highest-weighted combined candidate).

[0045] As indicated by block 255, the method 200 comprises selecting from the plurality of combined candidate summaries a first highest-weighted combined candidate summary and a second highest-weighted combined candidate summary. The candidates are weighted using one or a combination of factors, such as described in co-pending application (Attorney Docket No. GP-175-36-US), the entirety of which is incorporated herein by reference. In one embodiment, the weighting of the combined candidate summary is the sum of the weighting of the individual candidate summaries. Other formulae for combined weighting are possible, however, in alternate embodiments.

[0046] In another embodiment, the weighting is adjusted based at least in part on a percentage of keywords that are included within the candidate summary, so that a combined candidate which includes a greater number of the query keywords can have a higher adjusted weight than a combined candidate summary which includes a lesser number of query keywords. In another embodiment, the weighting is adjusted by a number of words in the combined candidate summary, so that candidate summaries with more words can have a higher weighting. In a further embodiment, the first and second highest-weighted combined candidates are further based, at least in part, on a percentage of keywords included in the plurality of keyword summaries.

[0047] As indicated by block 260, the method 200 comprises comparing the first and second highest-weighted combined candidates. The greater of the first and second highest-weighted combined candidate summaries generally represents an optimal summary of the document. In one embodiment, the method 200 comprises determining the optimal summary of the document based at least in part on the first and second attribute values. Of course, it will be appreciated that though the first and second highest-weighted combined candidate summaries generally will be the optimal choice from which to derive or obtain a summary of the document, other combined candidate summaries (i.e., not necessarily the first and/or the second highest-weighted combined candidates may be considered in alternate embodiments of the invention).

[0048] In another embodiment, the keyword comprises a first keyword and a second keyword. The optimal candidate summary may be based, at least in part, on a title of a document comprising the first keyword and one of the plurality of candidate summaries comprising the second keyword.

[0049] As indicated by block 265, the method 200 comprises constructing a summary result. In one embodiment, the summary result is the greater of the first and second highest-weighted combined candidates. In another embodiment, the summary result is a string comprising several of the highest-weighted first and second combined candidates. For example, the summary result can include the top three highest-weighted first and second combined candidates. In another embodiment, the summary result can include the three highest-weighted combined candidates, which together contain the greatest number of distinct query keywords. The summary result can be constructed using other suitable combinations of the combined candidates.

[0050] In one embodiment, summaries are displayed along with document titles, and the summary is computed independently of the title. In another embodiment, the summary is computed with consideration of the title. A size of the title and the summary may be adjusted accordingly. For example, there may be a fixed number of pixels available to display the title and summary. In such a case it may be desirable to shorten the title to display a longer summary. Candidate summaries may be generated from a title and the summary, and the optimal summary may be identified as above, and that the summary may begin with a candidate deriving from the title. The candidates for the title can start at the beginning of the title. There may be other information displayed along with the summary and title, such as the last access date of the document, or a URL, or filename identifying the document.

[0051] As indicated by block 270, the method 200 comprises communicating the summary result. In one embodiment, the summary result is communicated in a result set 124 to the client 102a via the network 106 as the summary of the document.

[0052] A computer readable medium of a server device, processor, or other device or application comprises instructions, that when executed, cause the server device, application, processor, or other device or application to perform method 200. Preferably, the server device, resource regulating application, and the computer readable medium are similar to that described above and with reference to Figure 1.

Alternatively, other suitable server devices, applications, computer readable media, processors, or other devices or applications can be used.

General

[0053] While the present invention has been disclosed with reference to certain embodiments, numerous modifications, alterations, and changes to the described embodiments are possible without departing from the sphere and scope of the present invention, as defined by the appended claims. Accordingly, it is intended that the present invention not be limited to the described embodiments, but that it has the full scope defined by the language of the following claims, and equivalents thereof.